Exploiting Bluetooth Low Energy Pairing Vulnerability in Telemedicine

Graduate Student Author: Wondimu K. Zegeye
Faculty Advisors: Dr. Richard A. Dean, Dr. Farzad Moazzami, Dr. Yacob Astatke
Department of Electrical and Computer Engineering
Morgan State University

Abstract

Telemetry has potentially large contributions to future medical applications. In the past decade wireless devices have invaded the medical area with a wide range of capability as components of a wireless personal area network (WPAN) and Wireless Body Area Network (WBAN). These applications in medical telemetry are not only improving the quality of life of patients and doctor-patient efficiency, but also enabling medical personnel to monitor patients remotely and give them timely health information, reminders, and support—potentially extending the reach of health care by making it available anywhere, anytime. This paper exploits the pairing vulnerability in Bluetooth Low energy (Bluetooth Smart) for HealthCare devices used in medical telemetry applications and demonstrates the key role security plays in telemetry.

Keywords: Bluetooth Low Energy, Security, Telemedicine

1 Introduction

Continuous and pervasive medical monitoring is now available with the presence of wireless healthcare systems and telemedicine services. Real-time health measurements are crucial in emergency situations. Treatment of a patient experiencing ventricular fibrillation within the first 12 minutes of cardiac arrest leads to a survival rate of 48%-75% according to the American Heart Association. The survival rate drops down to 2%-4% after 12 minutes have past [1]. With continuous medical wireless monitoring systems, patients’ information such as heart rate, blood pressure and electrocardiogram can be sent instantly to specialized medical centers located remotely to store and process properly. Medical emergencies can also be detected sooner and timely treatment can be applied. The effectiveness of Health care in several situations is improved significantly with current wireless communication technologies.

Wireless technology could be of great use without doubt for mass emergency situations like natural disasters or military conflict where patients previous records such medication examination history and other vital information are necessary. With the assistance of integrated wireless medical devices, the amount of time the doctors need to identify the problem, find out the medication history of the patient and consult fellow doctors located remotely will be reduced significantly. In addition, databases of patients built by continuous medical monitoring can be updated and accessed easily. As a result, the amount of paper work required will be reduced and the duplication of patient records will be avoided.
2 Background

Bluetooth low energy is designed to enable new markets requiring low latency, low cost, low duty cycle, low power consumption data devices. These markets include healthcare, proximity, fitness, automotive and smart grid applications. These may include device categories that are completely new to Bluetooth technology, or will use multiple features such as low energy and high speed Bluetooth technology within the same device.

The current evolution of Bluetooth standard is open and license free which can easily be integrated within existing Bluetooth technology. It creates an ideal environment for devices with very low battery capacity.

2.1 Bluetooth LE in Medical Devices

Embedding Bluetooth in several devices has been transforming medical devices for the last decade. The new era of medical devices are orders of magnitude better than their legacy counterparts, using software to extend or add its features and functions, especially to make the data more meaningful.

These new generations of medical devices are adaptable, multifunctional, portable, self-managing, self-documenting and intelligent. These features are possible since data processing and display are no longer limited to the device terminal or hardware. As these new smart medical devices are being incorporated into items that patients can carry around, wear, or even swallow, the opportunity of user interaction increases. This requires better presentation and usability of the information to drive appropriate engagement of the consumer in the self-health maintenance. In addition, we can communicate this information to anyone (e.g., physician) and anywhere (e.g., remote patient’s medical database system, etc.).

If we imagine the application of sensors in Healthcare which employ Bluetooth low energy, the possibilities and opportunities are diverse: continuous EEG monitoring with overlay of patient activities that would be triggers to epileptic activities (medical hat), 24 hour vital sign and ECG monitoring coupled with physical activity profile (medical underwear), simultaneous monitoring of airway resistance, environmental pollutants and actual medication delivery (medical inhaler), oximeter capable of adjusting the positive pressure to the changes in carbon dioxide concentration in the exhaled air of the user (medical CPAP), ingestible cameras capable of real time transmission of the images to image processing module, and many others. These devices offer additional data that can be used to create more interactive, and health monitoring an activity of daily living.

2.2 Bluetooth LE Pairing

The pairing mechanism is the process where the devices involved in communication exchange their identity information to set up trust and get the encryption keys ready for the future data exchange. Bluetooth has several options for pairing depending on the capability of the device and the user’s requirement.

In version 4.0 and 4.1 of the core specification, Bluetooth LE uses the Secure Simple Pairing model (SSP) (referred to as LE Legacy after the Bluetooth 4.2 release), in which devices based on the input/output capability of the devices choose one method from Just Works, Passkey Entry and
Although Bluetooth LE uses similar pairing method names to BR/EDR SSP, LE pairing does not use Elliptic Curve Diffie Hellman (ECDH)-based cryptography and provides no eavesdropping protection until release 4.2. Therefore, if an attacker can capture the LE pairing frames, s/he may be able to determine the resulting LTK. Because LE pairing key transport is used rather than key agreement, a key distribution step is required during the pairing.

Key generation in Bluetooth LE is performed by the Host on each LE device independent of any other LE device. However, the Key generation in BR/EDR is performed in the Controller. The key generation algorithms can be upgraded without the need to change the Controller, by performing key generation in the Host.

During Bluetooth LE secure connections, the following keys are exchanged between master and slave: Identity Resolving Key (IRK) for Device Identity and Privacy and Connection Signature Resolving Key (CSRK) for Authentication of unencrypted data.

In LE Secure Connections, the private/public key pair is generated in the Host and a Secure Connection Key is generated by combining contributions from each device involved in pairing.

Encryption in Bluetooth LE uses AES-CCM cryptography. The LE Controller will perform the encryption function. This function generates 128-bit encrypted data from a 128-bit key and 128-bit plaintext data using the AES-128-bit block cypher.

As shown in Figure 1, LE pairing begins with the two devices agreeing on a Temporary Key (TK), whose value depends on the pairing method being used. The devices then exchange random values and generate a Short Term Key (STK) based on these values and the TK. The link is then encrypted using the STK, which allows secure key distribution of the LTK, IRK, and CSRK.

\[1\text{ in [2] part 3.2.2, Figure 3-7. Bluetooth Low Energy Pairing}\]
2.3 LE Key Generation and Distribution

This subsection discusses the theoretical background behind the Key generation and distribution of Bluetooth LE as elaborated in [2].

Once the link is encrypted using the STK after phase II of Figure 1, the two devices distribute secret keys such as LTK, IRK, and CSRK. Two options are specified for key generation prior to distribution. A device may simply generate random 128-bit values and store them in a local database (called “Database Lookup” in the specification). The other option is to use a single 128-bit static but random value called Encryption Root (ER) along with a 16-bit Diversifier (DIV) unique to each trusted device to generate the keys. This option is called “Key Hierarchy” in the specification. For example, the keys can be derived from ER and DIV using the following formulas:

\[ LTK = d1(ER, DIV, 0) \]  \hspace{1cm} (1)

\[ CSRK = d1(ER, DIV, 1) \]  \hspace{1cm} (2)

\[ IRK = d1(IR, 1, 0) \]  \hspace{1cm} (3)

Where \( d1 \) is a called a Diversifying Function and is based on AES-128 encryption. However, the specification allows the user of other key derivation functions (e.g., those discussed in NIST SP 800-108, Recommendation for Key Derivation Using Pseudorandom Functions 14 ). The device does not need to store multiple 128-bit keys for each trusted device using this Key Hierarchy method; rather, it only needs to store its ER and the unique DIVs for each device. During reconnection, the remote device sends its DIV. The local device can then regenerate the LTK and/or CSRK from its ER and the passed DIV. If data encryption or signing is set up successfully, it is verified that the remote device had the correct LTK or CSRK. If unsuccessful, the link is dropped.
3 Exploiting Bluetooth LE Passkey Authentication

As described in [3], Bluetooth LE has certain cryptographic weakness of the passkey-based pairing which allows an attacker to impose an active attack. However, this paper is based on passive sniffing of packets during pairing which extends the scenarios that were already elaborated by Ryan [4]. The following subsections discuss the experimental setup used in this paper to perform the passive sniffing. By running the ubertooth-btle tool of the Ubertooth during the pairing between the LightBlue Bean and the Google Nexus 5 smartphone, Wireshark 1.12.4 is used to capture the packets during pairing.

3.1 Experimental Setup

![Experimental Setup Diagram](image)

Figure 2. Experimental Setup
The experimental setup used in this paper is as shown in Figure 2. It constitutes:

**Ubertooth One** The Ubertooth is a 2.4GHz transceiver built specifically to monitor and inject Bluetooth traffic. The Ubertooth project is open source, aimed at making Bluetooth security analysis available to anyone. All of the hardware plans and software needed to make one are available on the project's website [5].

**LightBlue Bean** The LightBlue Bean is a low energy Bluetooth Arduino microcontroller. Using Bluetooth 4.0, it is programmed wirelessly, runs on a coin cell battery, and is suited for smartphone controlled projects [6].

**Google Nexus 5 Smartphone** An Android 5.1.1 (Lollipop) based Smartphone with Bean Loader² application installed for wirelessly programming the LightBlue Bean.

**Laptop Computer** An IBM/Lenovo ThinkPad, Memory 1.5GB, Intel CPU T2600@2.16GHzX2, Ubuntu 14.04 LTS OS.

### 3.2 Results

Using a Wireshark with BLE header dissector capability such as version 1.10 and higher, visualizing the packets we have to make sure that we have the LL_ENC_REQ control PDU for the Crackle to work. Starting from that point, all the traffic is encrypted (Wireshark will report that as Bytes L2CAP Fragment). Running the crackle on the captured files we can get the clear file as in Figure 3. By applying the filtered captured BLE data, the crackle³ can guess or very quickly brute force calculate the TK (temporary key) used in the pairing modes supported by most devices (Just Works and 6-digit PIN). With this TK, crackle can derive the STK and LTK used during the encrypted session that immediately follows pairing. The LTK (long-term key) is typically exchanged in this encrypted session, and it is the one used to encrypt all future session keys between the master and slave.

In other words, the crackle works in such a way that, when the devices start pairing they exchange few values in plain text in the air that compromise the security. These values are confirm, rand, p1 and p2. Where confirm is a random value created for both devices generated by “Confirm value generation function” as specified in Bluetooth Core Spec. v4.0. This results in a passkey authentication commitment value of each device, rand is 128 bits long secret value whereas p1 and p2 represent known labels related to the actual public pairing parameters. Using a brute-force, the correct passkey TK (based on a 6-digit PIN) can be computed by trying all the values between 0 and 999,999, a million possible values. If the devices are using Just Works, the value of TK is always 0.

\[
confirm = AES(TK, AES(TK, rand \oplus p1) \oplus p2)
\] (4)

As can described in subsection 2.3, the secret key distribution in step III of Figure 1 can be clearly seen in 4 once the captured packets is decrypted.

As can be seen from Figure 5, the captured packets wireshark file ltk_exchange.pcapng file is passed as an input to the crackle tool and it gives ltk_exchange_clear.pcapng file as an output.


³crackle is open source software, available under the simplified BSD license in [7].
which is used for Figure 3. Once we get the TK and LTK, any future communications can be decrypted as follows:

```
$crackle -i <file_input.pcapng> -o <decrypted.pcapng> -l <LTK>
```

From Figure 3, MITM protection is set to 1 but the out-of-band (OOB) Flag is set to 0. This means that the eavesdropper would only need to try a million keys to get the correct user input pass key.

Better MITM protection is provided by using an out-of-band key set to 1. This would give a full 128-bit strength for encryption, and an eavesdropper would need to know the key to follow the conversation even if catching the whole pairing process.
In Bluetooth Core Specification version 4.2 release, security is greatly improved by the new LE Secure Connections pairing model. The numeric comparison method is added in addition to the other three methods (Just Works, 6-digit PIN and OOB) and the Elliptical Curve Diffie-Hellman (ECDH) algorithm is introduced for key exchange in this new model. But as described in [3], introducing D-H would not prevent any attacks which exploit the Flawed Bit Commitment of BTLE.

### 3.3 Comparision with Fitbit Flex

Using the Ubertooth-btle utility tool that comes as a part of the Ubertooth suite, we tried to exploit the device pairing process from Fitbit Flex to the Fitbit mobile application of the Google Nexus 5 using the set up in Figure 6. Crackle was unable to brute force the key exchange protocol. Similar previous works [8] demonstrate that Fitbit uses a different BTLE key exchange protocol namely ANT protocol. Even if Fitbit Flex has its own different vulnerabilities, its BTLE key exchange protocol provides it better protection than the vulnerability we demonstrated.
4 Conclusion and Future Work

This work highlights the potential for use of commercial technology such as Bluetooth in telemetry applications and demonstrates the critical need for good security for these to be used in sensitive applications. The use of wireless telemetry technologies such as Bluetooth Low Energy in medical environments brings major advantages to the existing HealthCare medical services. However, patients’ information must be secure and private, and be accessible only to authorized personnel. To ensure privacy of information, extra power and computation must be used to encrypt data to be transmitted. Guaranteeing information privacy and security is a critical requirement. This paper exploited the vulnerability of Bluetooth LE pairing passively and compares it with the Fitbit Flex pairing mechanism which uses an additional layer of encryption protocol. In addition, it shows how easily a classic BTLE pairing can be cracked using open source tools available and emphasizes the effect of this attack on the upcoming medical devices in Telemedicine.

In the future using this work as a basis, since most of the Bluetooth LE advertisement and data packets have the source addresses of the devices that are sending the data, third-party devices could associate these addresses to the identity of a user and track the user by that address. Then it is possible to track or localize Bluetooth LE capable devices. In addition, since the Bluetooth LE link with out a key exchange mechanism is a weak point to the end-to-end security for a telemedicine service, data stealing interception is possible.

References


[8] Britt Cyr et.al. ,“Security Analysis of Wearable Fitness Devices (Fitbit)”